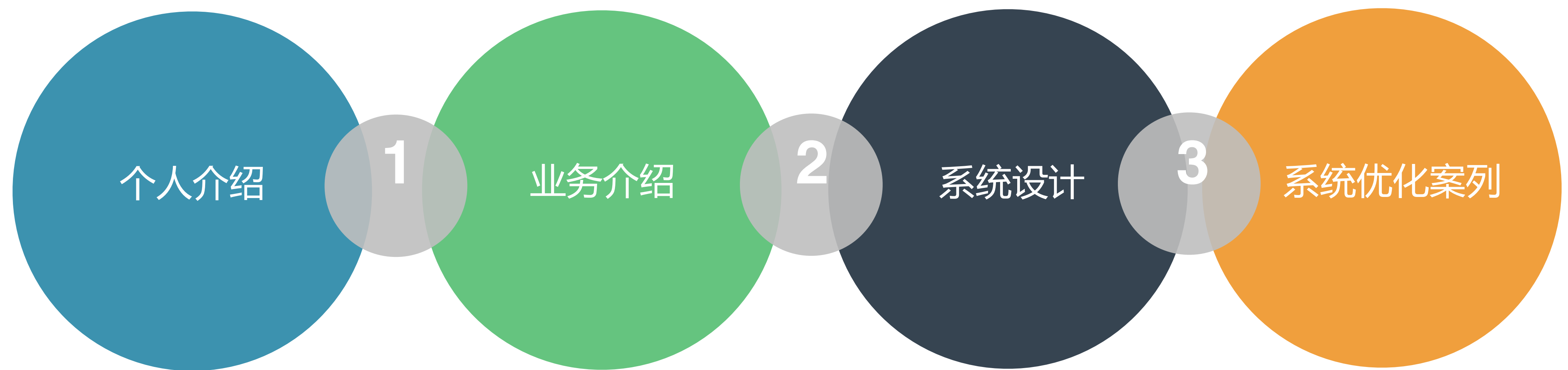


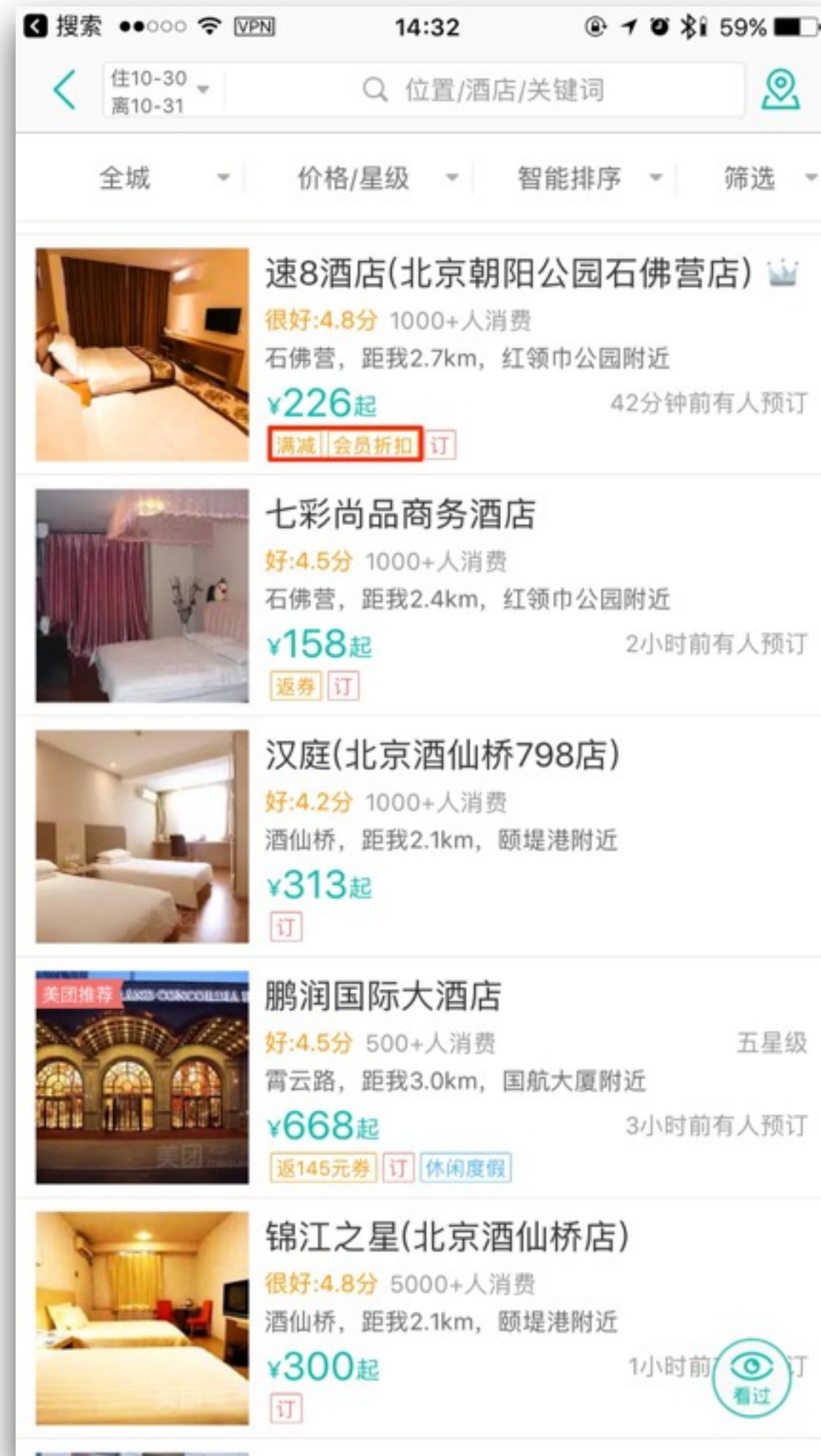
# 美团酒旅营销平台架构优化实践



# 个人介绍



# 业务介绍



if **this** then **that**

用户画像  
产品属性  
下单时间

.....

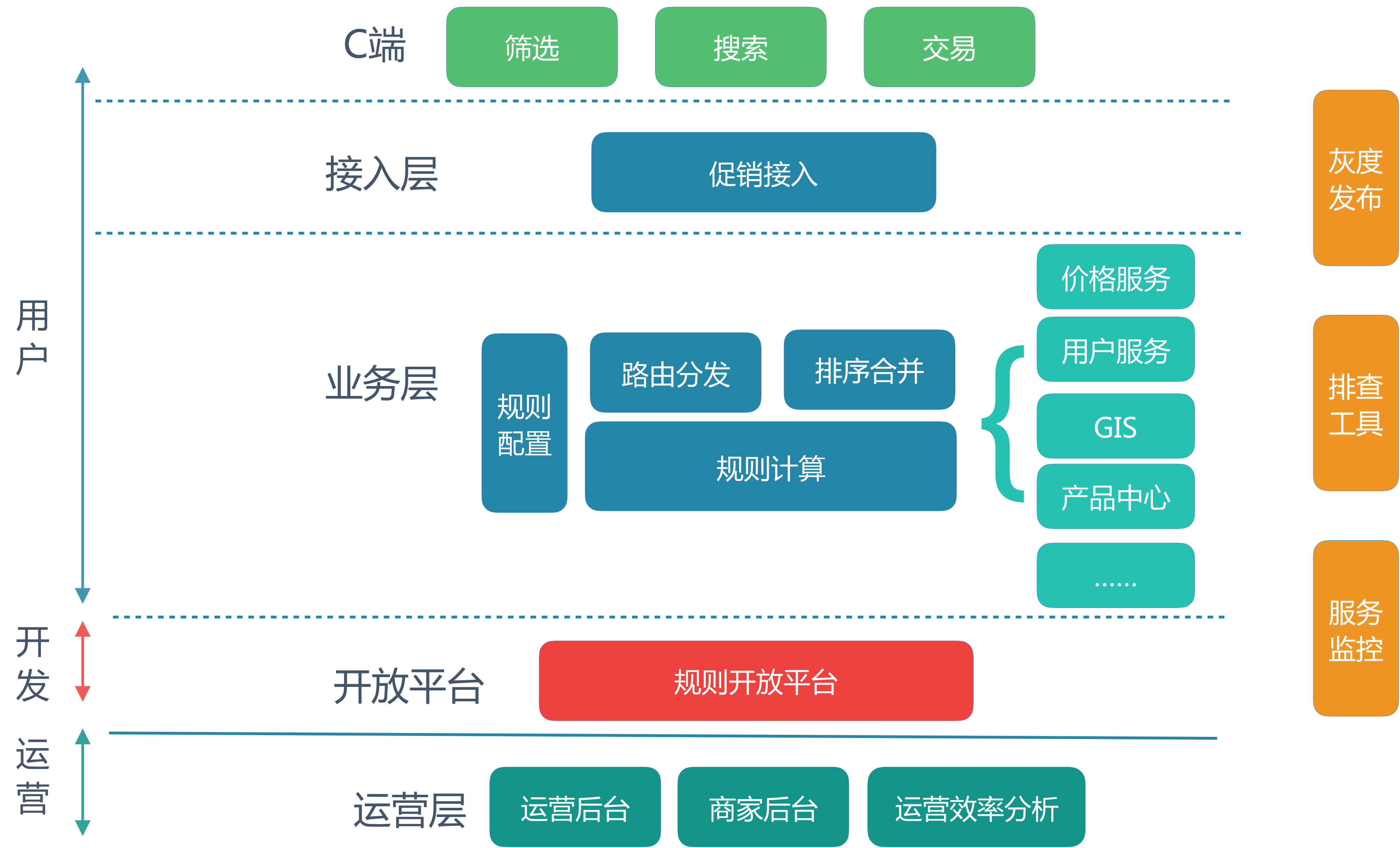
立减  
返券

# 业务介绍



承担全业务线流量  
支持业务快速发展

# 系统架构



# 挑战



## 性能

营销规则 200+  
活动 20000+  
QPS 50000+



## 模型

产品模型不一致  
待接入业务术语  
规则变更频繁



## 稳定

外部接口不稳定



## 扩展

扩容能力受限于数据库,缓存



## 效率

业务接入开发时间长



## 问题排查

营销规则复杂  
外部系统调用复杂



# 服务化

## 说起服务化时我们做什么



**功能拆分**  
前后端解耦  
服务内聚



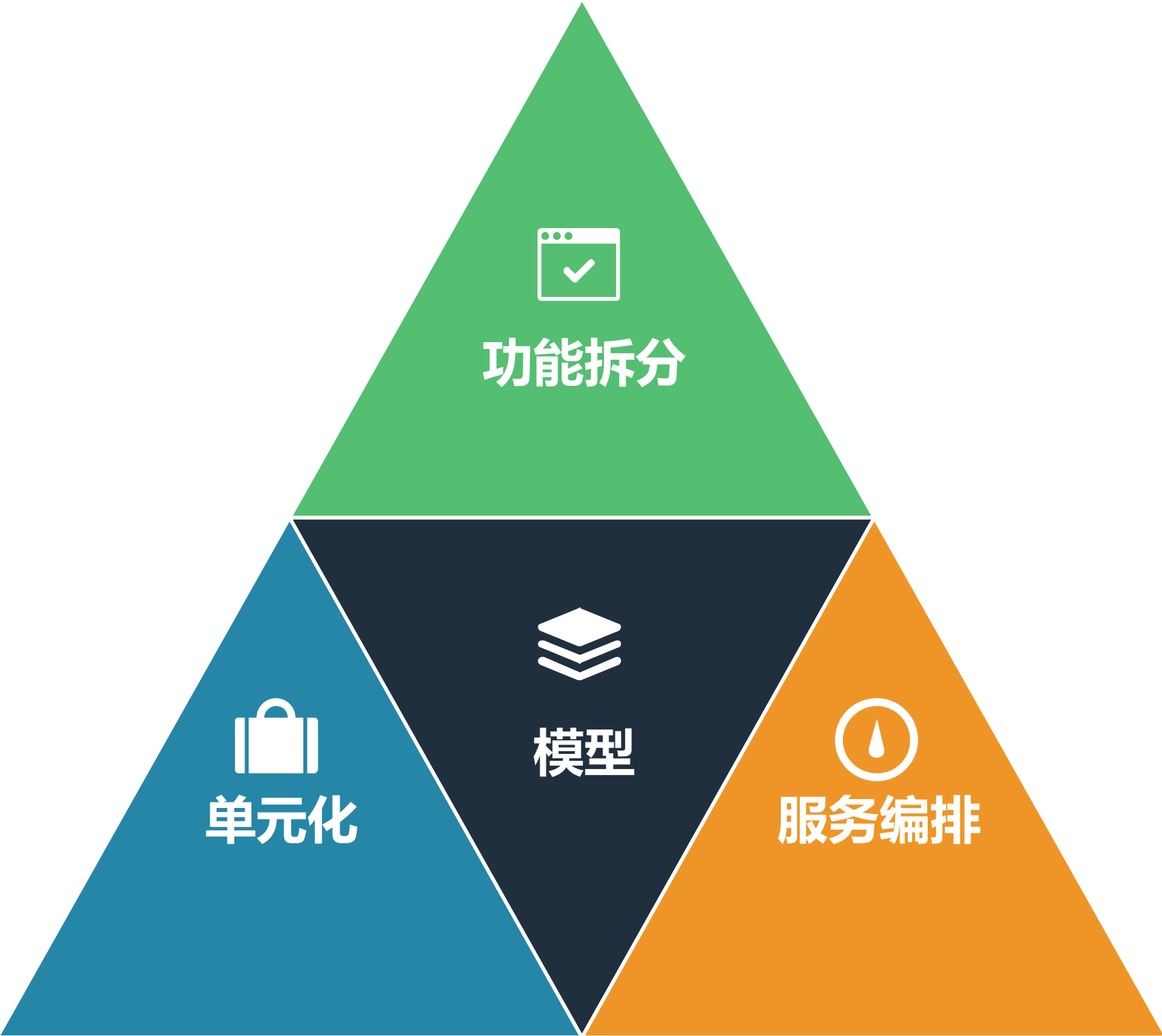
**服务编排**  
系统组合



**模型**  
正确的模型会带来简洁的实现  
业务模型 存储模型

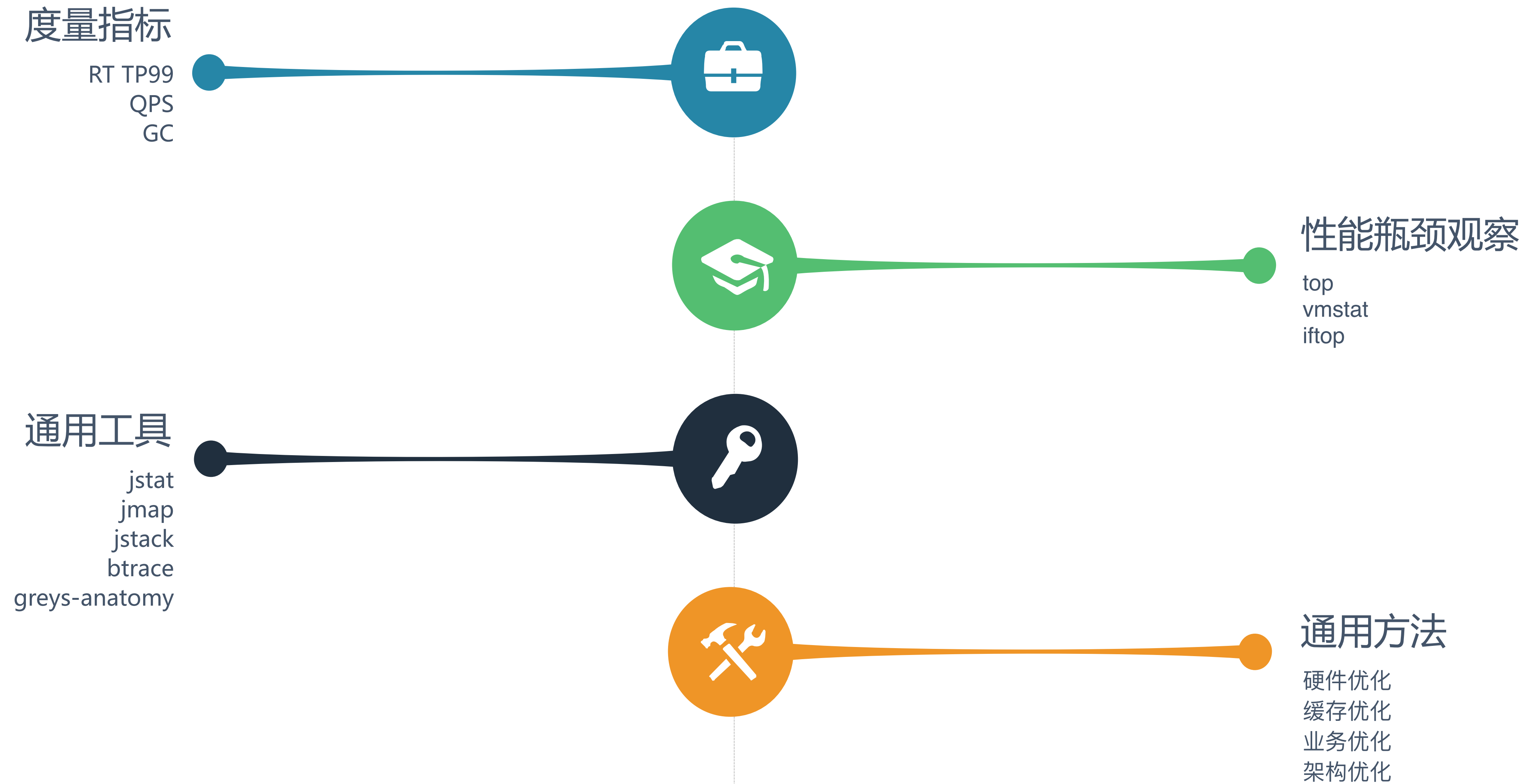


**单元**  
单元化部署 提升容灾能力


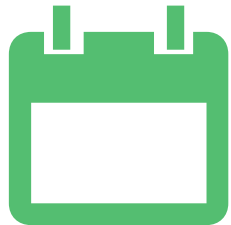






# 优化概述



# 场景分析

服务	 规则引擎	 选单服务	 运营后台	 开发平台
功能	运营规则匹配	产品ID映射服务	运营活动编辑	运营规则元信息编辑
热点	高	高	中	低
计算类型	CPU	CPU/IO	IO	IO
外部依赖	20+	1	5	0
模型复杂度	中	低	低	高

# 常见优化手段



# 缓存



读大于写



查询复杂度高



# 缓存

JVM是新的缓存

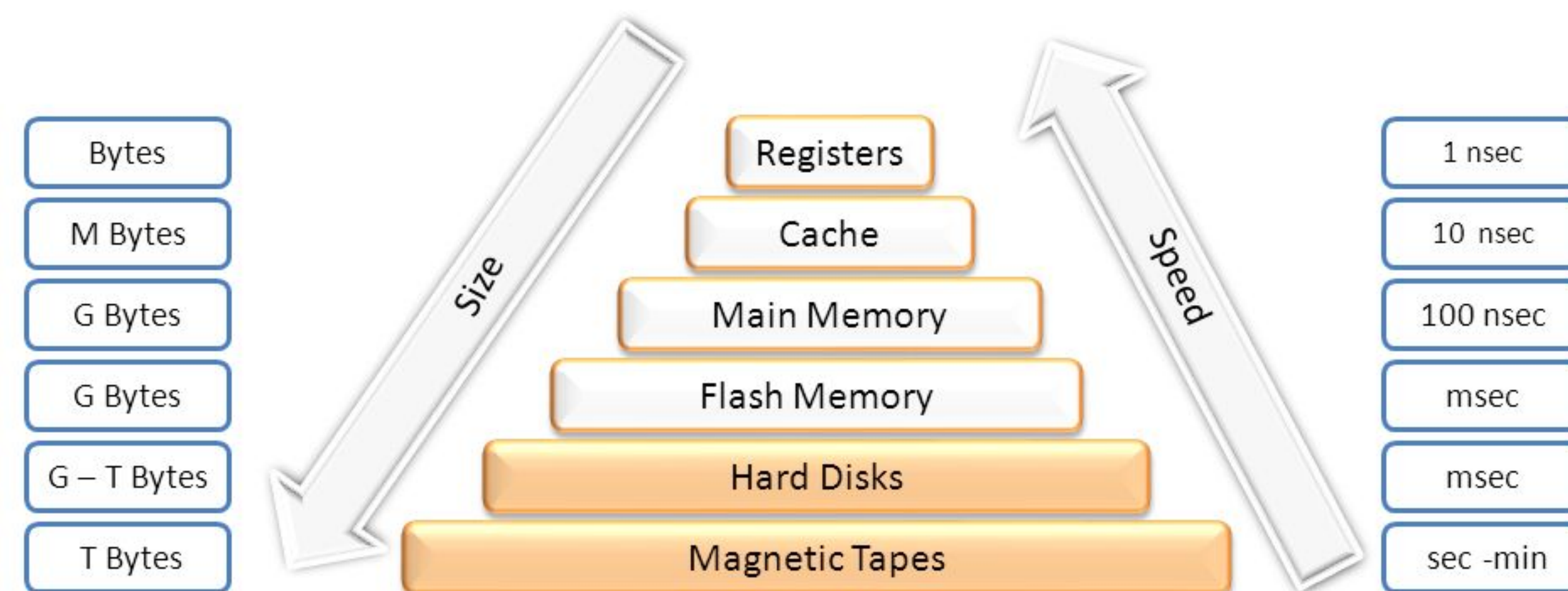


复杂的数据结构



减少IO 序列化成本

## Storage Device Hierarchy



**Hard Disk** and **Tapes** have mechanical movements  
**Others** do not have and called Solid State Devices (SSD)

# 案例

## 促销引擎系统拆分



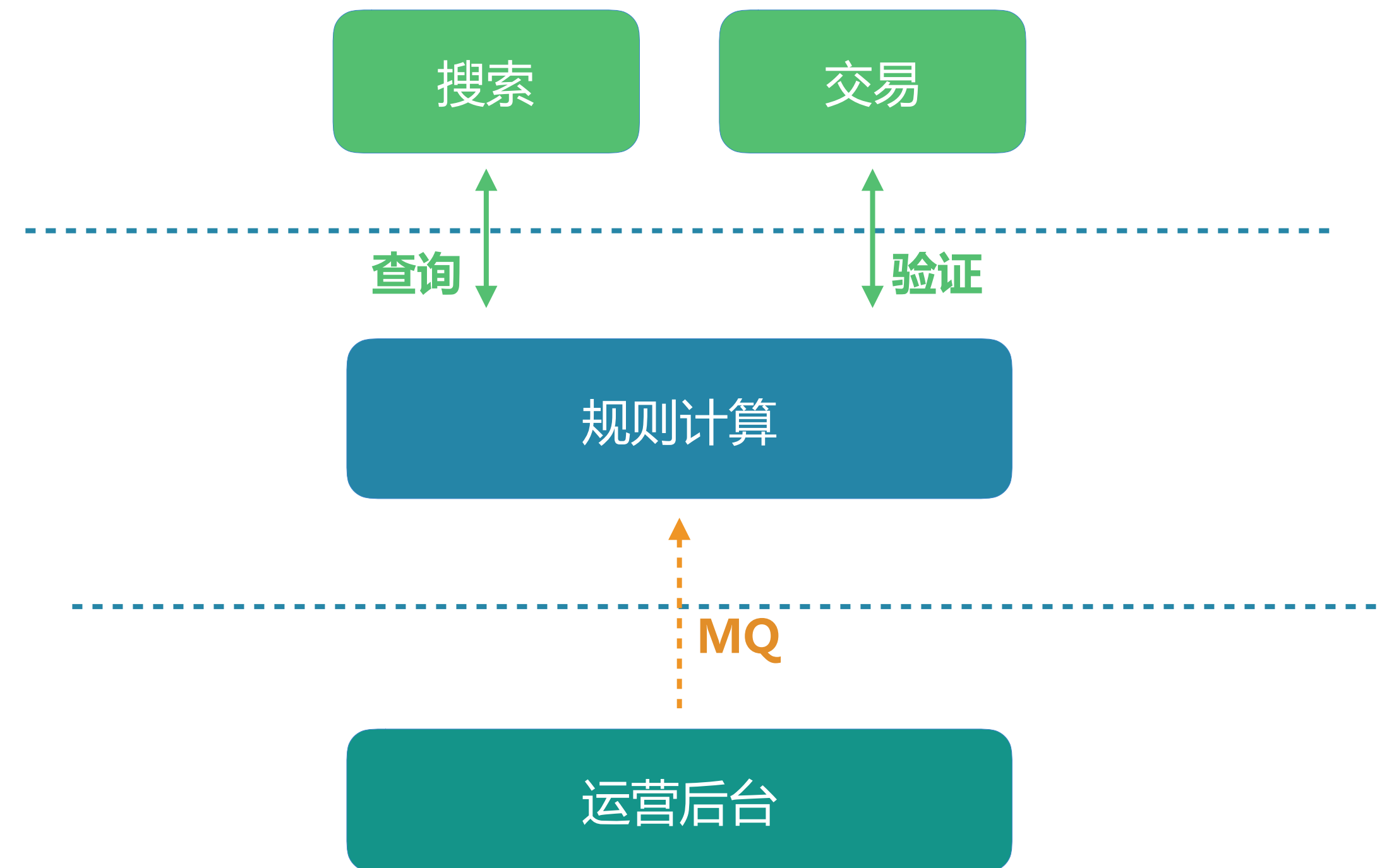
### 读写分离

- 使用MQ异步通知变更
- IO密集与计算密集应用独立部署



### 数据存储

- 规则信息存储于jvm中
- 避免请求到DB中
- 服务无状态 可水平扩展



# 模型

## 业务映射数学模型



简化表达

区间型

价格



50

≤ 值 <

80

元

离散型

所选入住城市

不包含



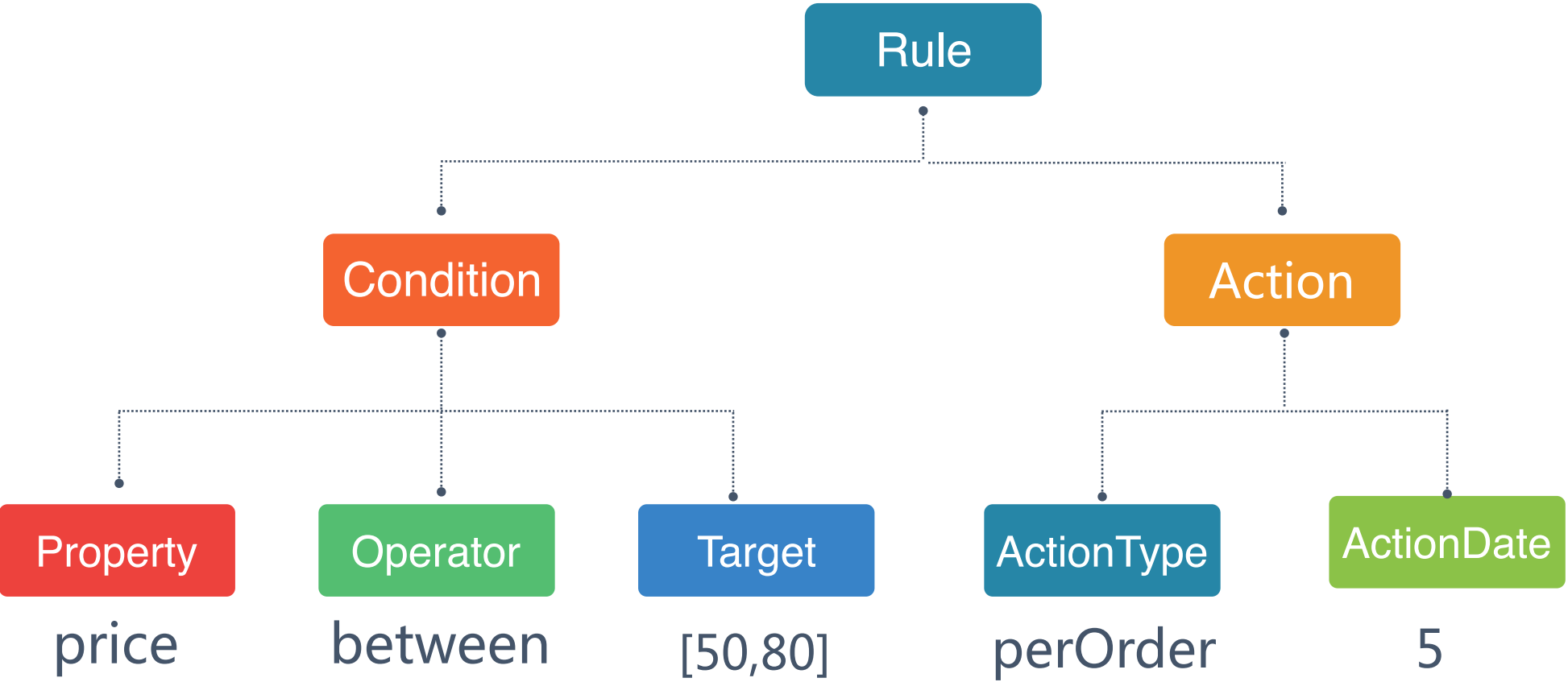
厦门市



实现一致性约束

# 模型

if 价格 50 ≤ 值 < 80 元 then 每订单立减 5 元





# 模型

潘多拉规则管理平台

Search...

Q

ROOT

部门

产品类型

促销类型

索引类型

PPRelation

运算符相关

Target相关

属性相关

Action相关

条件相关

规则相关

规则

规则模板

规则

新增

ID:

ID

部门:

输入名称搜索

产品类型:

输入名称搜索

Action:

输入名称搜索

名称:

名称

描述:

描述

更新时间:

操作人:

操作人

Q 查找

🗑 清空

🔄 刷新

ID	部 门	产 品 类 型	Action	条件组	名称	描述	创建时间	更新时间	操作人	
1	1	1	1	<div>酒店预付-酒店范围</div> <div>酒店预付-用户范围</div> <div>酒店预付-产品范围</div>	立减活动规则	立减活动规则	2016-06-24 18:17:32	2016-07-07 14:04:15		<div>查看</div> <div>编辑</div> <div>删除</div>
2	1	1	1	<div>酒店预付-订单</div> <div>酒店预付-用户范围</div>	酒店预付满减规则	满减活动规则	2016-07-04 15:05:35	2016-07-07 14:08:42	daimao	<div>查看</div> <div>编辑</div> <div>删除</div>
3	1	1	2	<div>酒店预付-用户范围</div>	返抵用券规则	返抵用券	2016-07-04 15:14:05	2016-07-04 15:14:05	daimao	<div>查看</div> <div>编辑</div> <div>删除</div>
4	1	1	2	<div>酒店预付-订单</div>	满赠抵用券规则	满赠抵用券	2016-07-04 15:16:09	2016-07-04 15:16:09	daimao	<div>查看</div> <div>编辑</div> <div>删除</div>
5	1	1	9	<div>酒店预付-酒店范围</div> <div>酒店预付-用户范围</div> <div>酒店预付-产品范围</div>	折扣活动规则	折扣活动	2016-07-04 15:17:28	2016-07-07 14:57:35	daimao	<div>查看</div> <div>编辑</div> <div>删除</div>

# 算法

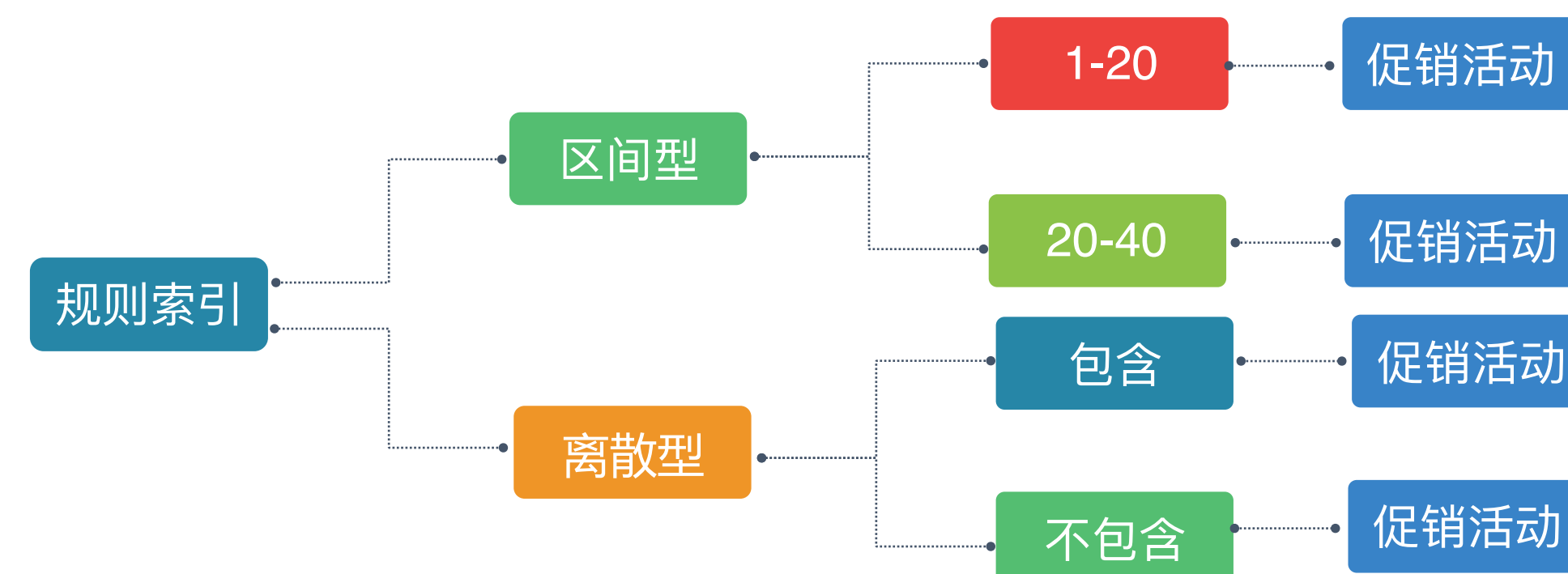
## 问题

- 计算量大：200 规则 \* 20000活动 \* 500 deal  
= 20亿
- GC 频繁：每分钟8次
- CPU使用率高：CPU 使用率超过70%

## 索引结构

区间型 价格 50 ≤ 值 < 80 元

离散型 所选入住城市 不包含 厦门市



## 算法模型



# 运维

```
@HteyeInvoker(description = "价格计算",paramDesc = "checkIn&checkOut&poi\" ,\"分割\")
public Map<Integer, Map<Integer, PriceResult>> calc(Date checkIn, Date checkOut, String poi)
```

方法调用 日志级别调整 内存监控 开关设定 添加机器 帮助

选择机器列表 10.32.141.224:8412(dx-travel-ruleengine21)

在线活动详情 ⓘ

initContditionIndexService.getActiveInfos(int)

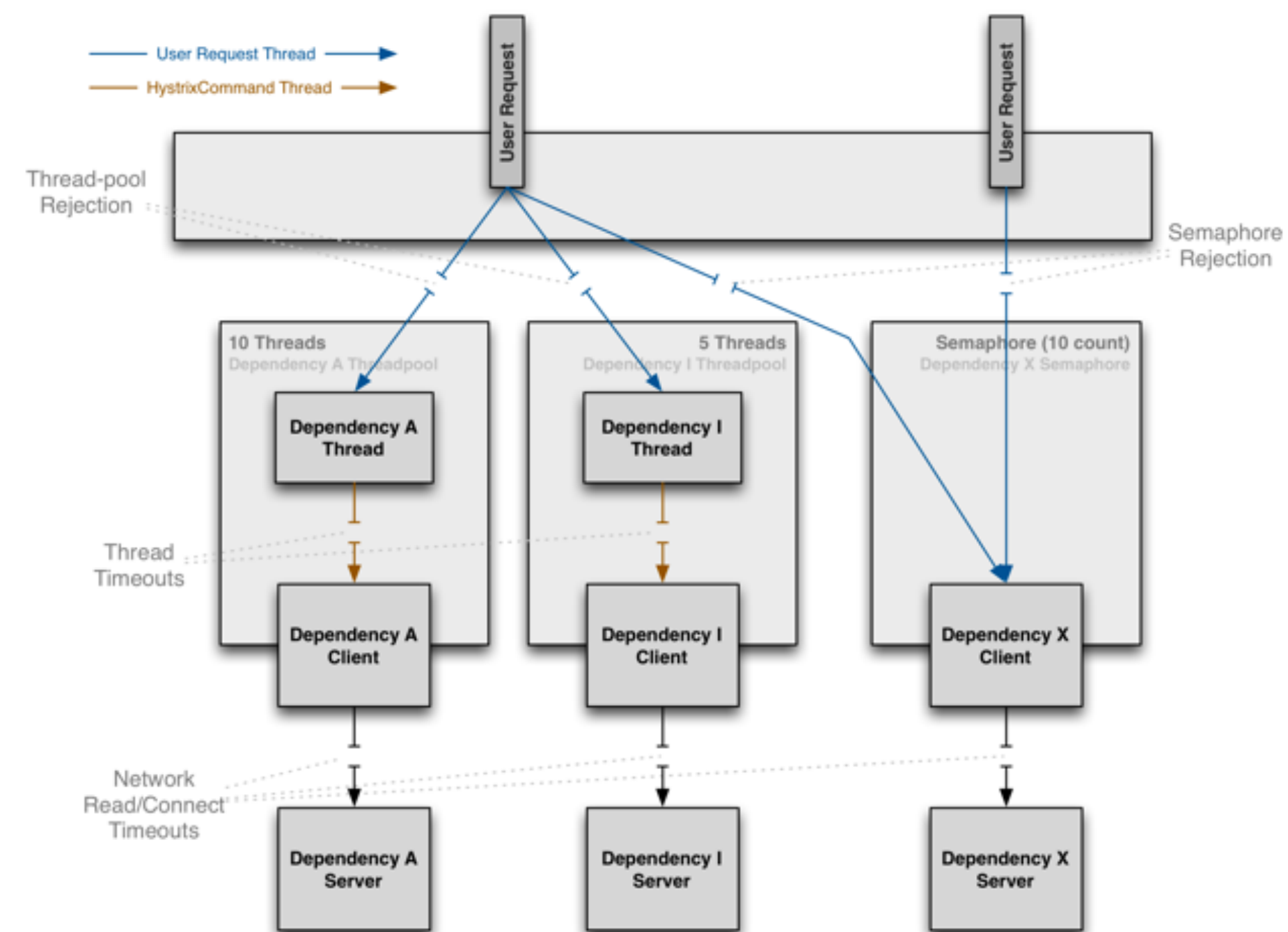
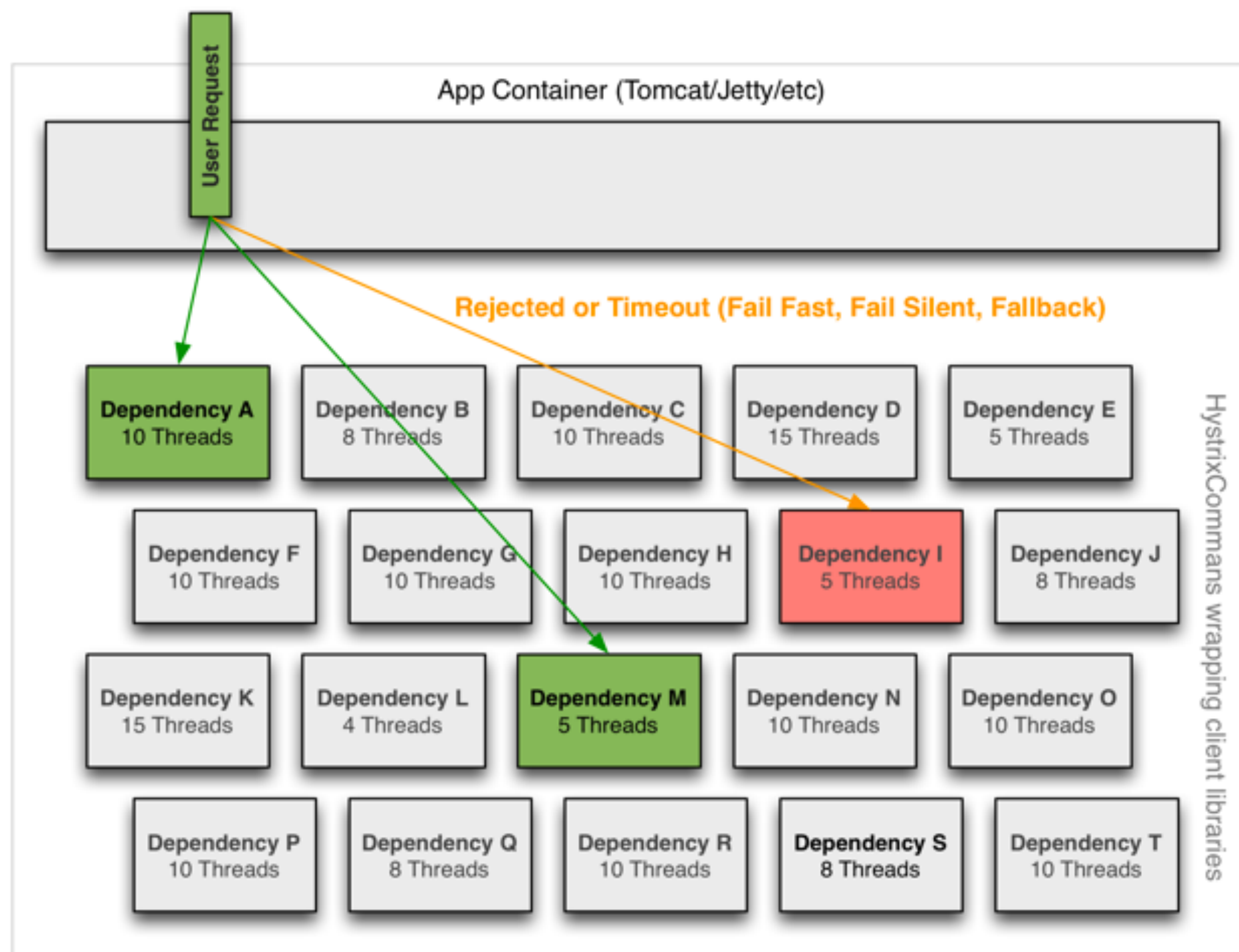
type int

call method

方法调用 日志级别调整 内存监控 开关设定 添加机器 帮助

选择机器列表 10.32.141.224:8412(dx-travel-ruleengine21)

logger	level	action
root	DEBUG	DEBUG ⚙
ROOT	ERROR	ERROR ⚙
java.sql.PreparedStatement	ERROR	ERROR ⚙
java.sql.Statement	ERROR	ERROR ⚙
msg	INFO	INFO ⚙
rule_engine_handle	ERROR	ERROR ⚙
rule_engine_init	INFO	INFO ⚙
rule_error	ERROR	ERROR ⚙





# 总结



## 数据结构

- 业务映射到数学模型上
- 简化业务逻辑的表达
- 实现了优化和一致性约束



## 算法

- 分析问题确定本质
- 倒过来看问题



## 架构

- 读写分离 提升性能和资源使用率
- 全内存架构 扩容及时
- 服务化 提升复用性和可用性



## 抽象

- 提取业务共性 建立规则模型
- 平台化业务共建



## 工具

- 线上方法调用 日志级别调整 内存占用查看
- Hystrix组件 隔断不稳定依赖

